Advances in Science, Technology and Engineering Systems Journal
Vol. 6, No. 2, XX-YY (2022)
www.astesj.com

ASTES Journal
ISSN: 2415-6698

Profiling Attack on WiFi-based IoT Devices Using an Eavesdropping of an Encrypted Data Frames

Ibrahim Alwhbi Alharbi[*,1], Ali Jaber Almalki[2],Mnassar Alyami[3],Cliff Zou[4],Yan Solihin[5]

[1]*L3Harris Engineering Center, Computer Science, University of Central Florida, Orlando, 32816 , USA*

[2]*Bisha, Computer Science and Information Technology, University of Bisha, Bisha, 67714, Saudi Arabia*

[3]*L3Harris Engineering Center, Computer Science, University of Central Florida, Orlando, 32816 , USA*

[4]*L3Harris Engineering Center, Computer Science, University of Central Florida, Orlando, 32816 , USA*

[5]*L3Harris Engineering Center, Computer Science, University of Central Florida, Orlando, 32816 , USA*

ARTICLE INFO

Article history:
Received:
Accepted:
Online:

Keywords:
Internet of Things
Privacy Attack
Eavesdropping

ABSTRACT

The rapid advancement of the Internet of Things (IoT) is distinguished by heterogeneous technologies that provide cutting-edge services across a range of application domains. However, by eavesdropping on encrypted WiFi network traffic, attackers can infer private information such as the types and working status of IoT devices in a business or residential home. Moreover, since attackers do not need to join a WiFi network, such a privacy attack is very easy for attackers to conduct while at the same time invisible and leaving no trace to the network owner. In this paper, we extend our preliminary work originally presented at the CCNC'22 conference by using a new set of time series monitored WiFi data frames with extended machine learning algorithms. We instrument a testbed of 10 IoT devices and conduct a detailed evaluation using multiple machine learning techniques for fingerprinting, achieving high accuracy up to 95% in identifying what IoT devices exist and their working status. Compared with our previous work in , the new approach could achieve IoT device profiling much quicker while maintaining the same level of classification accuracy. Moreover, the experimental results show that outside intruders can significantly harm the IoT devices without joining a WiFi network and can launch the attack within a minimum time without leaving any detectable footprints.

*[1] *This paper is an extended version of the paper published in Alyami, Mnassar, Ibrahim Alharbi, Cliff Zou, Yan Solihin, and Karl Ackerman. "WiFi-based IoT Devices Profiling Attack based on Eavesdropping of Encrypted WiFi Traffic." In 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), pp. 385-392. IEEE, 2022.*

1 Introduction

The emerging smart infrastructures are integrated with the Internet of Things (IoT) devices and their applications to make daily life easier for individuals and improve the public environment [2]. To this end, IEEE 802.11 Wireless network (WiFi) is a significant development that helps connect a wide variety of IoT devices such as smartphones, smart TV, home automation, intelligent vehicles, surveillance cameras, health monitoring, and many more [3]. The increase in applications increases the attention of attackers that find the loopholes and gain maximum knowledge of users' private information. The connected devices, digital systems, and sensors that play a vital role in people's daily life cause a significant threat of privacy leakage of private information [4]. For example, the Mirai malware attack, which triggered distributed denial-of-service (DDoS), generates attacks on WiFi-connected IoT devices and applications [5]. Additionally, worms in smart bulbs gave attackers access to all adjacent IoT lights that were compatible [6]. To this end, the infrastructure and IoT applications must incorporate privacy protection during intelligent network design and development phases. Figure 1 shows an overview of IoT ecosystems and relevant scenarios.

Considering the aforementioned privacy threat, this paper focuses on the attacker's ability to fingerprint the IoT devices in the

[*] Corresponding Author: Ibrahim Alwhbi Alharbi, UCF Orlando, FL, & ia@knights.ucf.edu

https://dx.doi.org/10.25046/aj050501

WiFi network. The existing work mostly assumes the attacker is inside the network where the attacker has to either join the network prior to fingerprinting or wiretap the network link of the WiFi network [7, 8, 9]. This assumption cannot be satisfied in the real world by most attackers as most WiFi networks have secured password protection, and very hard to impossible for attackers to have physical access to their WiFi routers/access points.

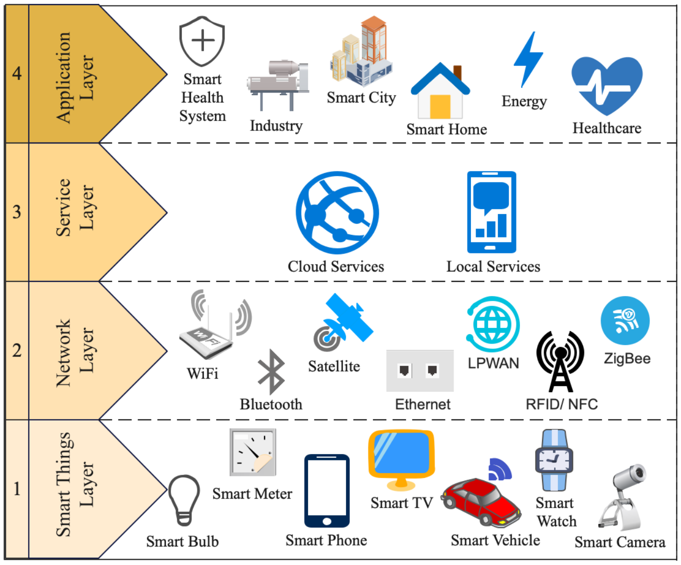


Figure 1: The overview of IoT ecosystem and applications scenarios.

To this end, in this paper, we conduct a brief investigation of whether an outside attacker can identify a network's IoT devices without having a joined in a WiFi network. Compared to the entities in the network, the outsider attacker may face several difficulties, i.e., the attack may be prevented from analyzing the plaintext of packet payload due to the WiFi data-link layer encryption. On the other hand, the outside attacker captures the fingerprint of the data packet, which would be noisy (especially when more nodes are present in the network). Moreover, the captured data cannot obtain the IP address or the port information; thus, it is difficult to assume whether the hypothesis is correct or not.

However, if the hypothesis is correct, there could be major threats as follows:

- The attack is possible on all WiFi networks which are in close proximity to the attacker. So it is easy to attack those with weak passwords.

- The attacker only needs to drive or walk a short distance to begin analyzing traffic, where no specific preparations are required beforehand.

- The attack is usually untraceable because it leaves no traces. Thus it is undetectable neither for users nor for forensic investigators.

The hypothesis needs to be investigated in detail based on the above points.

To this end, we demonstrate that the outside attacker cannot only fingerprint the IoT devices, but it is a straightforward process that one can achieve depth information about the network devices. The significant contributions of the proposed research are summarized as follows:

- We conduct a detailed investigation and prove that fingerprinting the IoT devices from the outside of the network eavesdropping is not only feasible, but it's a straightforward process.

- We consider nine real-world IoT devices and capture the out-of-network WiFi traffic in two modes, idle and active, using the sniffing tool capable of single-channel and multi-channel monitoring.

- We explore the time-series data and train a machine-learning algorithm to profile nine real-world IoT devices and present their prediction accuracy. The experimental results prove that the fingerprint can be possible by achieving high accuracy of up to 95%.

The rest of the paper is organized as follows: In section 2, we present a brief literature review that motivates us to conduct this research work. In section 3, we first define the problem statement and then present the threat model and assumptions of the proposed research. Section 4 briefly explains a detailed procedure for capturing WiFi traffic from outside the network. In section 5, we conduct data processing and analysis for profiling attacks based on machine learning. Section 6 shows the experimentation and proves that the hypothesis above is true. Finally, in section 7, we conclude the proposed research and define the future work.

2 Related Work

Thanks to the IEEE 802.11 protocol and the development of Wi-Fi-enabled devices, instant access to the internet is now possible everywhere near a public AP through a Wi-Fi connection [10]. Billions of people's lives have been significantly impacted by this development throughout the world. However, when a massive number of people are involved, there are higher chances of misuse or exploitation [11]. The general public can now be followed and profiled due to security vulnerabilities brought on by the ease of use of the 802.11 protocol suite and the number of open public Wi-Fi hotspots [12].

Since the early days of the internet, device identification has been one of the primary targets for classifying network traffic [13]. Several studies have been conducted on WiFi and ethernet, proving that various information can be extracted from IoT devices using traffic classification, such as the device type and the device's activities [14]. The existing literature predicts that a collection of TCP/IP level packets would allow observation of network activity. Those techniques are used to extract the device's potential information. In [15], the authors claim the possibility of a single attribute signature for a variety of IoT devices by employing a port number. Moreover, the deep learning technique is used to perform device fingerprinting for flow volume features. However, this technique is not suitable for the network which is accessed from the outside adversary because the IP traffic is encapsulated in the upper layer that encrypts all significant network features such as cipher suites, protocol, and port number. To this end, we employ a unique collection of attributes that are straightforward to extract the features even from outside the network. In [16], the authors used hardware fingerprinting and extracts clock skew measurement. This major focus in this study is more on the hardware rather than the device-specific classification which is considered in the proposed research. In [7], the authors used traffic analysis of WiFi, Bluetooth, and ZigBee to identify the status of the devices and proposed a defensive strategy based on traffic spoofing. In order to gain the traces of WiFi, the authors use a rogue access point with tcpdump, which means the authors assume that all the adversaries are either a part of the network or the adversary already contains significant knowledge about the network. However, the proposed technique considered that the adversary doesn't have any prior knowledge, and the attack is conducted from the outside of the network. There are many other studies that have been conducted with a focus on WiFi traffic analytics from the outside of the network where the traffic is analyzed using off-the-shelf monitoring devices [17]. However, to the best of our knowledge, none of the existing studies reports the missing rate using those off-the-shelf sniffers or the lost frames as a result of channel hopping eavesdropping [18].

On the other hand, several studies have been conducted on the defense mechanism for those traffic analysis attacks [19, 20]. However, the primary focus of the existing literature is either on location anonymity or website fingerprinting. Moreover, the techniques such as padding or traffic morphing result in less accuracy of the classifier [21]. Besides, those methods cannot rely on time-based classification because of their limited capabilities for obfuscating traffic patterns.

In contrast to comparable work mentioned above, the proposed research presents an alleged privacy attack against WiFi-based devices that rely on WiFi traffic monitoring from outside the network. We proceed with a precise and practical proof-of-concept attack on the assumption of a realistic threat scenario. We also discuss a possible defense against those attacks.

3 Problem Statement, Threat Model, and Assumptions

In this section, we first define the problem statement that serves as a strong motivation for the proposed research. Afterward, we present the threat model that needs to be covered by the proposed study. Finally, we discuss the considered assumptions while conducting the proposed research.

3.1 Problem Statement

The TCP/IP paradigm is used for communicating devices on networks. The IP address at the Network Layer and the MAC address at the Data Link Layer can be used to identify the devices on the network. Spoofing identities have been used to get around these identifying mechanisms and access restricted resources. Using WiFi traffic analysis, an attacker can "fingerprint" devices to determine private user behavior [22]. For instance, by continuously watching the camera's bitrate, the attacker could ascertain the movements of objects inside a building [23]. Moreover, the attacker can predict which vulnerabilities are available to exploit depending on the type of IoT devices in the network. However, extensive research has been conducted on fingerprinting the IoT devices using eavesdropping from the inside network [24]. This research performs a detailed investigation and proves that fingerprinting the IoT devices eavesdropping from outside the network is not only possible but also a straightforward process. The developers of the IoT devices must need to consider some extra security constraints to overcome those privacy threats.

3.2 Threat Model

In the proposed research, we consider that the attacker aims to target the information of IoT devices using a targeted WiFi network. Moreover, the attacker is also interested in the number and type of unique IoT devices, e.g., Laptop, Smart TV, Light Bulb, etc. Moreover, the attacker is also interested in getting the mode of those devices, such as idle or active. The attacker intends to gain maximum sensitive information by gathering the devices' data. For example, the device type may reveal potential vulnerabilities to software/hardware status. The number of devices may reveal the customers in business, the number of employees, or the family size. The type and number both can reveal the status of socioeconomic.

Considering this as a potential threat model, we aim to perform a detailed investigation that fingerprinting the IoT devices eavesdropping from outside the network is a straightforward process. This threat should be considered in the first place.

3.3 Assumptions

In the proposed research investigation, we assume that the attacker continuously observes the network traffic outside the network using the targeted WiFi or access point. We also believe that the attacker is physically in the signal range of the access point, so he can perform eavesdropping using a sniffing tool and gather the nearby WiFi network traffic. We assume that the attacker can't join or break the network. To this end, in the proposed research investigation, we prove that fingerprinting the IoT devices from the outside of the network eavesdropping is possible. Moreover, the existing research focuses on the IoT devices operated at 2.4GHz; we consider the same. However, the proposed study can be applied to 5GHz as well.

4 System Model

In this section, we first present the system architecture, and then we discuss how the attacker captures the network traffic from outside of the network. After that, we present the pre-processing of the captured data.

4.1 System Architecture

In the proposed investigation, we consider the system architecture illustrated in Figure 2 where the attacker uses to access the WiFi network. The system architecture consists of two stages, offline and online. The first stage (offline) is the attacker's profiling model training and building stage, where an attacker uses his computer and many IoT devices to conduct experiments in order to build the

https://dx.doi.org/10.25046/aj050501

profiling model of each IoT device. On the other hand, the second stage (online) is the attacking stage, where the attacker monitors a WiFi network, trying to identify all IoT devices in the WiFi network based on monitored data and profiling models built in the offline stage. In the first stage (offline), the attacker configures maximum IoT devices which are connected to the nearby WiFi gateway. The attacker accesses the network traffic using a sniffing tool, where the traffic data would be labeled as the device name using the MAC address. The collected data is then pre-processed, where we removed the noise (e.g., network traffic gathered from nearby WiFi networks, data link layer broadcast frames, WiFi protocol beacon frames), and dumped the valuable features into a CSV file for applying the machine learning techniques. In particular, we apply several machine learning algorithms and achieve accuracy up to 95% for device identification.

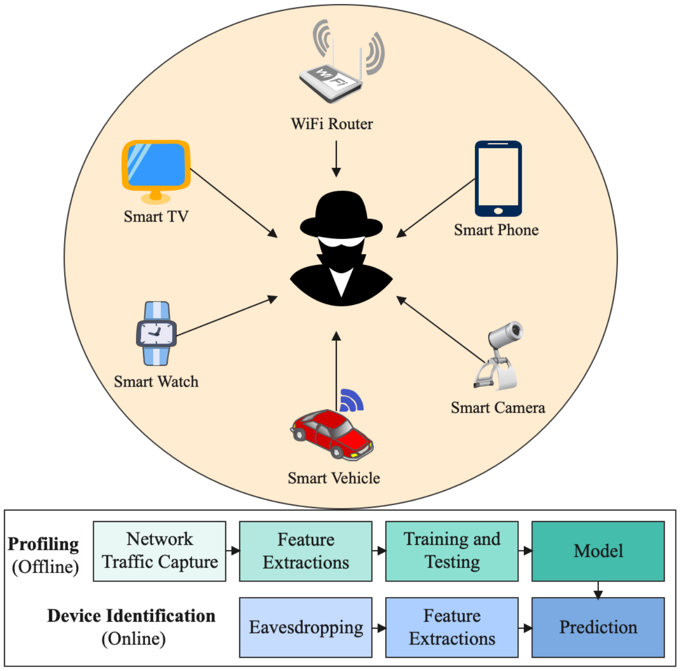


Figure 2: The attacker's perspective of profiling attack on the IoT devices.

In the second stage (online), the attacker applies a sniffing tool and targets the victim's access point for a short period of, for example, 30 seconds, and stores the traces for pre-processing. To this point, we never require prior knowledge of the IoT devices for pre-processing, which we will explain in detail later in the following subsection. Precisely, we use standard and statistical filtering techniques to eliminate the noise from the data frames which do not represent the patterns of data. After that, we use Python scripting to extract the features from pre-processed data. Finally, we were able to predict the type of devices and their activity.

4.2 *Traffic Capturing From Outside of the WiFi Network*

In order to capture the data frames from outside of the WiFi network, we first test to use the two most popular sniffing tools, Kismet[1] and Airodump-ng[2]. Kismet stores the network traces as SQLite3 database, whereas Ariodump-ng dump the traces into a capture file format such as pcap[3]. The output of pcap is used to perform packet inspection as the output is in a compatible format, where the packet inspection can be done via a network analyzer such as Wireshark[4]. We use those sniffing tools because of their capability to sniff raw 802.11 frames. Besides, both of them are able to monitor single-channel and multi-channel using frequency hopping. For the hardware, we use an external wireless adapter (Alfa AWUS036ACM) as the built-in WiFi cards don't serve our purpose because they are programmed to accept the data packets which are particularly addressed to the machine's interface card.

Once the captured traffic is analyzed, we observe that a significant proportion of captured packets contradicts the elephant-mouse internet traffic phenomenon [25]. The elephant flows of 1500 bytes were unable to see for all the available devices, including the video packets captured from a Camera or a smart TV. The proposed investigation shows that the aforementioned tools can only capture a limited range of packets in terms of their sizes. For example, they are able to capture the packet up to the frame size of 472 bytes; however, this size is enough for particular applications such as signal intelligence. Due to such limitations of Kismet and Airodump-ng, we consider another sniffing tool called Airtool[5]. The Airtool sniffer is a MAC's built-in sniffing tool that can passively sniff WiFi traffic and store the traces in a pcap format which can be further analyzed using Wireshark. We simultaneously run Wireshark on a different laptop connected to the network to record its own incoming/outgoing network traffic to the AP in order to confirm the accuracy of the traffic caught by Airtool which is running on an out-of-network MacBook. The traffic between the second laptop connected to the network and the AP was considered for comparison of those two traces.

Size Range of Packets/Frames	Wireshark	Airtool	
	#Packets	#Frames	#Data Frames
0-19	0	2428	0
20-39	0	5593	199
40-79	2441	0	0
80-159	260	2890	2883
160-319	108	239	239
320-639	173	194	190
640-1279	241	255	255
1280-2559	13574	13846	13846
Total	16797	25445	17612

Table 1: Comparison of captures by out-of-network Airtool with the in-network Wireshark.

Because Airtool also records additional control and management frames at the data-link layer (most frames have sizes between 0 and

[1] https://www.kismetwireless.net
[2] https://aircrack-ng.org/doku.php?id=airodump-ng
[3] https://en.wikipedia.org/wiki/Pcap
[4] https://www.wireshark.org/
[5] https://www.intuitibits.com/products/airtool

39), which are absent from Wireshark's in-network traffic capturing, we discovered that Airtool collects more frames than packets recorded by Wireshark, as shown in Table 1. Every frame that is collected by Airtool is translated into a WiFi data-link layer frame, whereas every frame that is captured by Wireshark is translated into an Ethernet II frame. As a result, for the same WiFi packet, the Wireshark capture is smaller than the Airtool interpretation of the data-link layer frame. This explains why the 2441 packets in the Wireshark capture that are between 40 and 79 bytes all show up in the Airtool capture's higher packet size range (80 to 159 bytes). Additionally, the comparison holds even after excluding all control and management frames from the Airtool capture (as seen in the last column), so there aren't any apparent missing packets according to Airtool. As a result, we employ Airtool as our testbed for evaluation.

4.3 Data Pre-Processing on Captured Data

Once the encrypted WiFi traffic data is captured using the Airtool software, the output in pcap format is analyzed using the Wireshark tool. In particular, the following steps are taken to analyze the captured data:

1. We start with the traffic broadcasting in both directions to the MAC address of the WiFi network under investigation. This is required since Airtool could potentially monitor WiFi traffic from many neighboring APs. Only data frame types are kept since all other control, and management MAC-layer frames do not adequately depict the profiling data pattern.

2. We export the pcap files into the csv files for the following steps number 3 and 4.

3. We eliminate noisy frames that some MACs produced. Since they often only appeared as a single frame, these noise frames are simple to filter out. By just keeping traffic frames with bi-directional communication traffic, they are filtered away.

4. To make dataset labeling easier, we swap out the MAC addresses for the relevant device names and their operational status. This step is included only in the offline training stage.

5. The dataset required for both offline training and performance testing is eventually obtained using a Python script that extracts and calculates statistical characteristics.

5 Data Processing and Analysis

In the section, we first discuss the observable Data Fields, and then we discuss data analysis. Finally, we discuss device profiling on time-series data.

5.1 Useful Data from Network Monitoring

As we have discussed earlier, the process of out-of-network monitoring. Here, we present the observable data fields on a secured WiFi network where everything above the data-link layer is encrypted because of WiFi protocol *WPA-PSK*. Because of this encryption, the only observable data is the MAC-layer frame header, signal strength, and the observation timestamp. The frame header provides the source and destination's MAC addresses, frame size, and frame type. However, the signal strength cannot be used as it has been affected by several factors such as neighboring WiFi networks, deflection, absorption, and reflections of the surrounding objects. Therefore, in the proposed investigation, we neglect the signal strength and only use the MAC layer frame header.

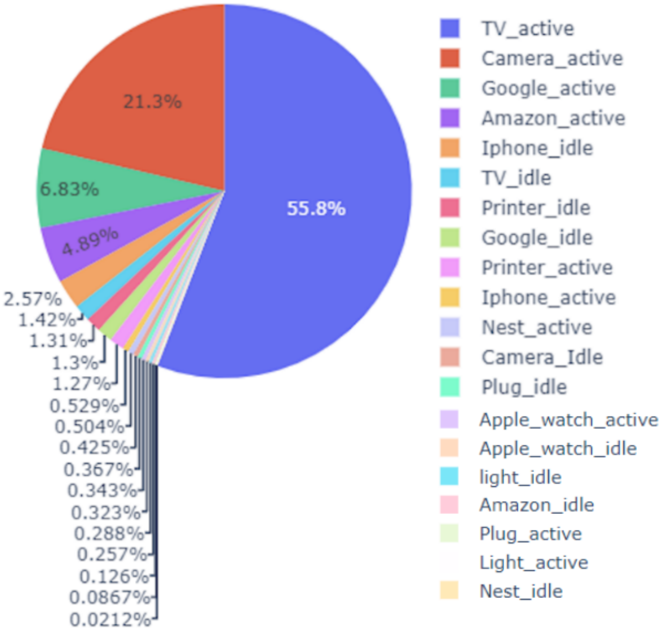


Figure 3: The total number of data packets captured with respect to the IoT device. Many devices' names show whether they are in active or in an idle state.

5.2 Data Analysis

As discussed earlier, we consider 10 different IoT devices and collected their data through out-of-network monitoring. In Figure 3, we show the total percentage of captured data from each device. To provide insights into the monitored traffic, we measure the working and idle status of all the IoT devices. From Figure 3, we can observe that the number of received packets from TV is more than the other IoT devices such as the camera and Google. All of the other devices' captured data are comparatively way less than those three devices. Consequently, the camera and google both display different behavior with regard to packet sizes, as shown in Figure 4. Specifically, the camera and google both appear to send the majority of their packets at a fixed size of 170 bytes and 140 bytes, respectively, as shown in Figure 5. We will discuss this part in the following section.

As shown in Figure 3, we believe that the attacker can easily build the signature. Moreover, the attacker can also change the status of those IoT devices. As we can see in Figure 3, there is a huge difference between the active and idle states of the devices which means the AP initiated communication to send an off signal to those devices. On the other hand, due to the notable decline in flow when switching to the idle state, the working condition of other devices with better network capabilities and memory storage, such as iPhones, printers, and Amazon, is impressively noticeable. For instance, Amazon will only get a small number of packets when it is idle because the user is not searching the Internet. Similarly, we show a detailed transmission of data packets from each IoT device in Figure 4. In particular, we show the total number of packets sent from the access point to the IoT device, which is represented by **1**, and the total number of packets sent from IoT devices to the access point, which is represented by **0**. Moreover, the figure depicts both the total data captured in the active and idle states of all the devices.

5.3 *Machine Learning Techniques*

In order to execute our investigation, we choose several classification methods, and a popular machine learning model XGBoost [26]. We consider XGBoost because of its superior performance, especially for the problems of network classification among other popular machine learning models. Because our data is captured in two different sequence sizes, so we consider precision, recall f-1 score, and support vector machine (SVM) as performance metrics to tackle the time series data.

5.4 *Profiling on IoT Devices using Time-series Data*

Processing time-series data is simple. The captured traces from the Data-link layer is converted into a string of three-feature items. The monitored frame is then converted into three numeric values: the size of packet P, the direction of packet X, and the arrival time Y, where **0** represents the transmitted packets and **1** represents the received packets by an IoT device. Following those numeric values, we can obtain the series of data such as $\{P_0, X_0, Y_0\}, \{P_1, X_1, Y_1\}, ..., \{P_n, X_n, Y_n\}$. Figure 5 shows the heatmap of the correlations for each feature in the dataset, where **1** shows the maximum value, and **0** shows the minimum.

This method's main disadvantage is that it needs a lot of packets to supply all the data points needed for categorization or machine learning training. In our case, we are dealing with a heterogeneous system monitored inside a specific time window; certain devices (such as Smart TV) generate a large volume of data packets while others only produce very sparse packets (such as smart light bulb). For instance, if we compare how long it takes the light and TV to collect a 100-packet series, the TV just needs one second of visible data while the light needs approximately 30 minutes.

To overcome this challenge, we use a two-level categorization technique, starting with a traffic intensity threshold. Devices are divided into two groups in the first level according to whether there is a high or low volume of traffic. Then, in accordance with the volume of device traffic, we use an appropriate sequence size. Using ML algorithms, the second level determines the prediction probability. A prediction is made if the probability rises above a certain threshold; else, the data is labeled as an ”unknown” device. Once the dataset is created, we extracted the following features from the monitored traffic in each time window:

- Packets transmitted and received by the access point to and from the IoT devices, respectively.
- The difference in inter-arrival time.
- Total number of bytes in the transmitted and received packets.
- Variance of sizes in transmitted and received packets.
- Variance of size distribution in transmitted and received packets.
- Mode of transmitted and received packets.
- The average number of consecutively transmitted or received packets before seeing a received or transmitted packet, respectively.

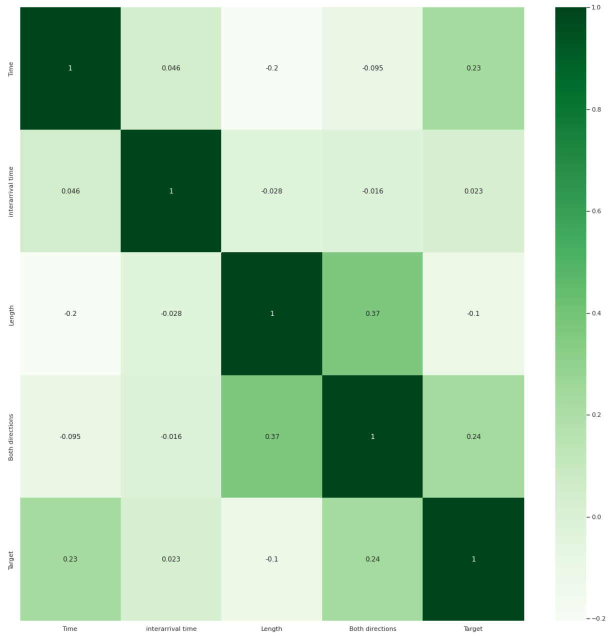


Figure 5: Correlation of features in a dataset.

6 Results and Discussion

In this section, we first present the testbed settings and the evaluation metrics. Afterward, we show the IoT devices' packets transmission and their reception in terms of packet length, and time. Finally, we show the evaluation results and prove that the outsider intruder can significantly harm the IoT devices without joining the WiFi network.

6.1 *Testbed and Evaluation Metrics*

With the help of a WiFi router, we built up a testbed with 10 distinct IoT devices. We use AirTool to capture the WiFi data frames between all IoT devices and the WiFi router for an appropriate amount of time in order to collect enough data. Once the data is captured, we use a time-series format and randomly split the dataset into two groups, 20% for testing and 80% for training.

The following metrics are used to assess our classification models: Precision, Recall, F1 Score, and Accuracy. Let's use the abbreviation T to stand for true prediction, further subdivided into true positives and true negatives. The letters F stand for false prediction, which is further divided into false positives and false negatives. The following equations are used to measure the Precision, Recall, F1 Score, and Accuracy, respectively.

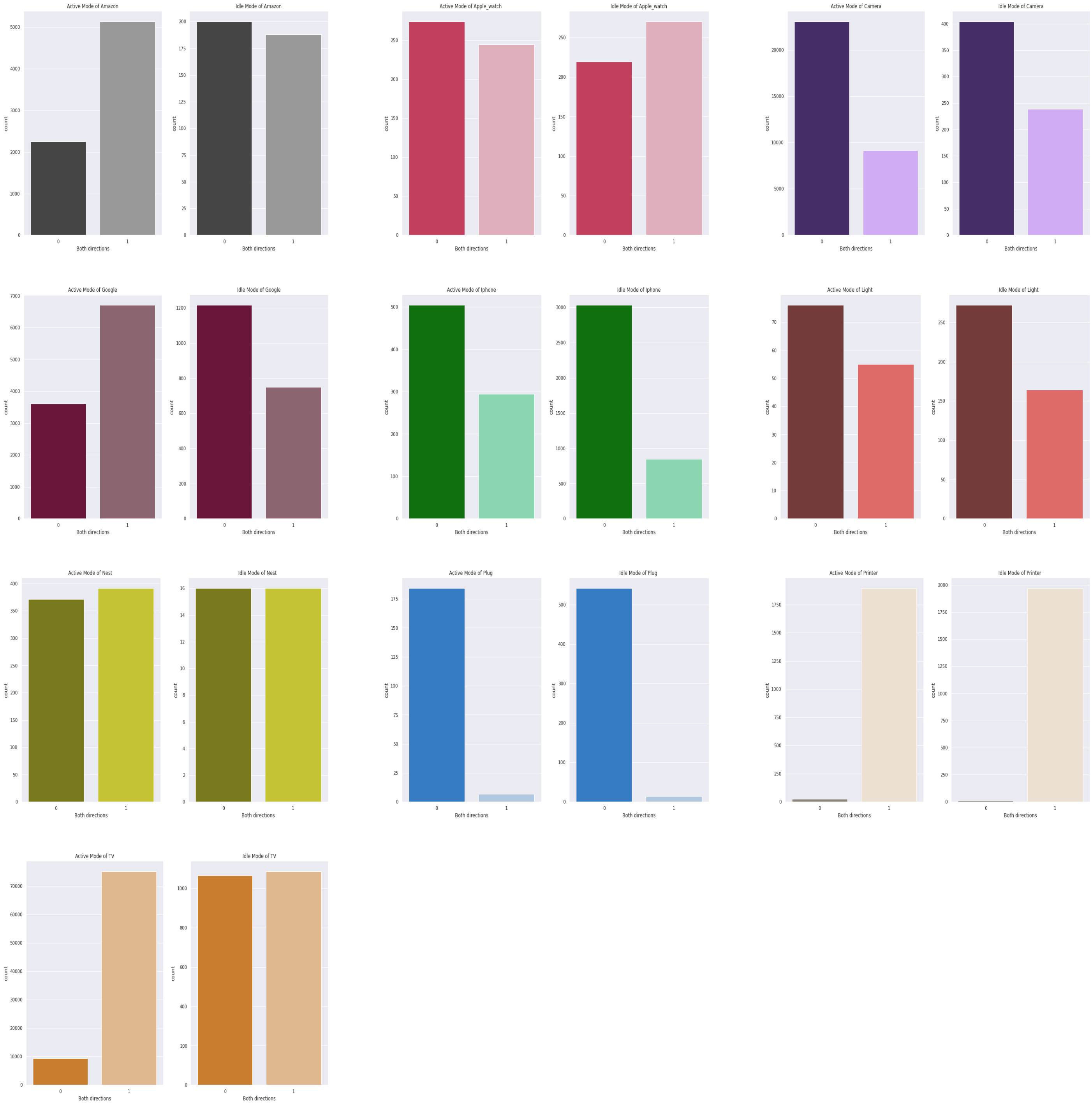
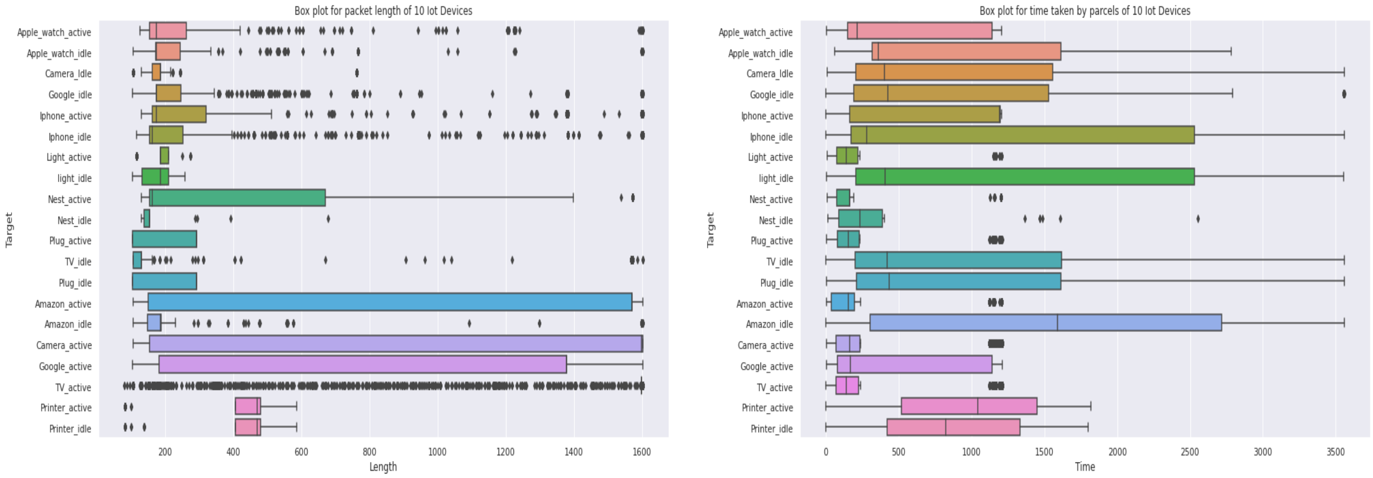


Figure 4: Total number of packet transmission from access point and IoT devices, respectively. In particular, the figure shows the total number of packets sent from the access point to the IoT device, which is represented by **1**, and the total number of packets sent from IoT devices to the access point, which is represented by **0**.



(a) Box plot for packet length frames of 10 IoT devices.

(b) Box plot for the time taken by packets of 10 IoT devices1.

Figure 6: Packet length and time-taken in transmission or reception of 10 IoT devices.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

$$Accuracy = \frac{T}{T + N} \tag{4}$$

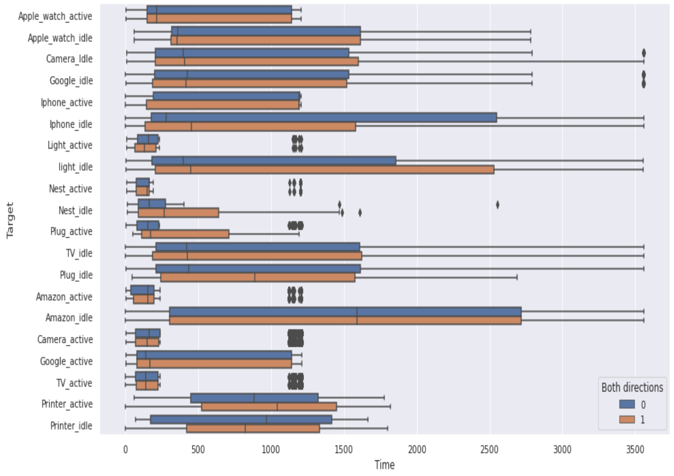


Figure 7: Packets transmission in both directions of each IoT device with respect to time.

6.2 Captured Data

Before going to the evaluation, we first show the captured data in terms of packet length and time. In Figure 6, we show the packet length and time taken in transmission or reception of 10 IoT devices, respectively. In particular, in Figure 6 (a), we show a box plot for the packet length of 10 IoT devices. The packet length of each device is captured in both active and idle states. Similarly, in Figure 6 (b), we show a box plot for the time taken by packets to transmit or receive by each IoT device in both active and idle states. To further show the significance of captured data, in Figure 7, we show the packet transmission in both directions (IoT device to access point and access point to IoT device) with respect to time. The line in the middle of the box shows the average length of the packet transmitted or received. In particular, **0** represents the packets transmitted from the IoT device to the access point, and **1** represents the packets transmitted from the access point to the IoT device. The above-mentioned box plots prove that a significant amount of data is captured in the proposed investigations, where an attacker can easily access the useful information of the IoT devices and can significantly harm the user. We also prove that the IoT devices can reveal the potential vulnerabilities to hardware/software status as the attacker can change the status of IoT devices.

Considering this as a potential achievement in our investigation, below we present the results of model accuracy, which can further support our claim.

6.3 Results

We chose a 30-minute time window size for evaluation. We believe that long-term observed traces can teach us more about time-series patterns than short-term ones, which call for much longer time observations to carry out the attack. As we were facing a challenge in working with imbalanced data, for example, data captured from some devices are extremely high such as TV or Camera, whereas other devices are barely showing any record for example Nest, Light as shown in Figure 3. To balance such data, we use SMOTE analysis to prove the significance of the results. Figure 8 shows the confusion matrix of prediction accuracy using a SMOTE analysis on the XGBoost model. The Figure shows that the IoT device "*Printer_active*" captures maximum true labels, whereas the IoT device "*google_idle*" captures minimum true labels.

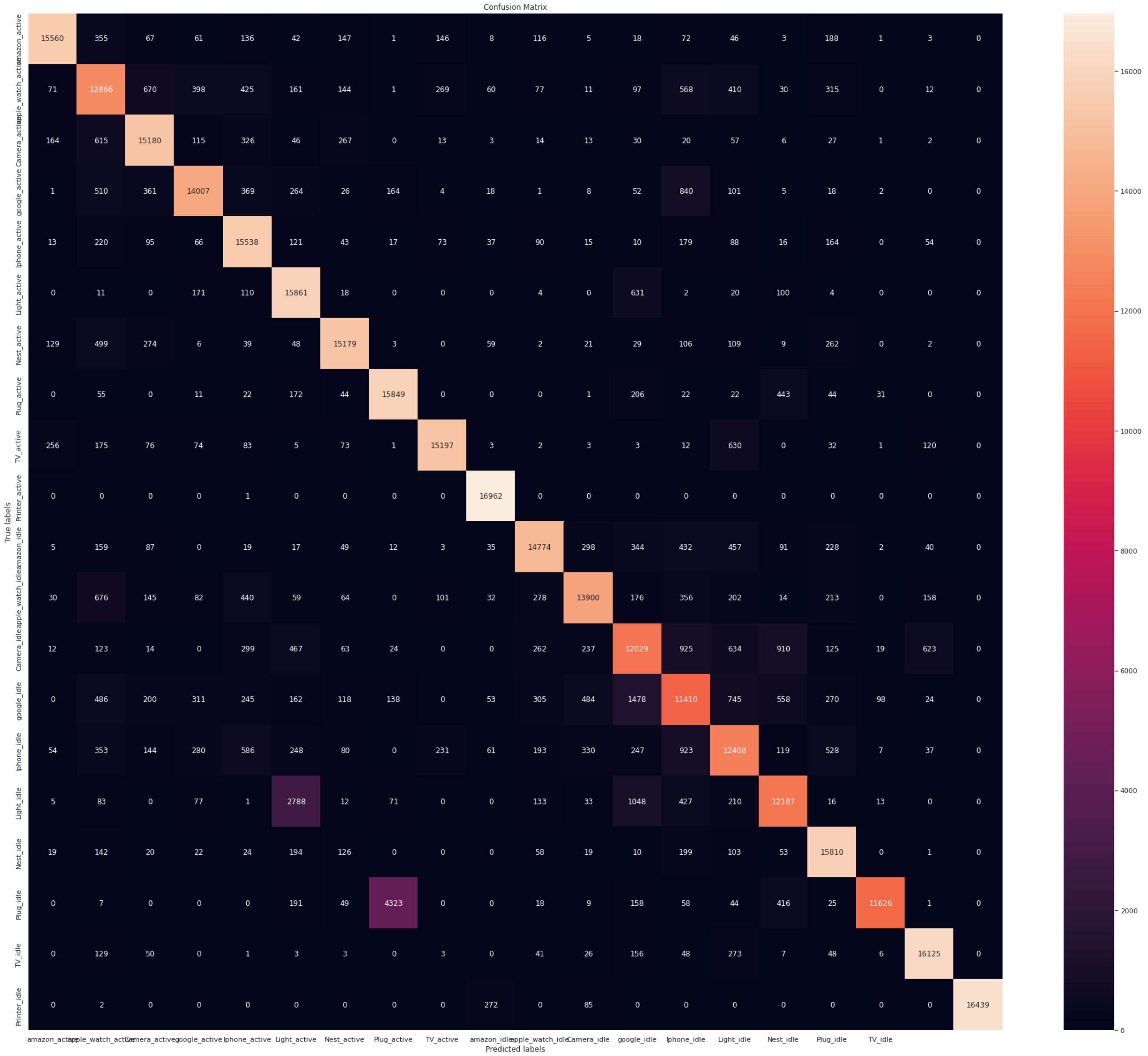
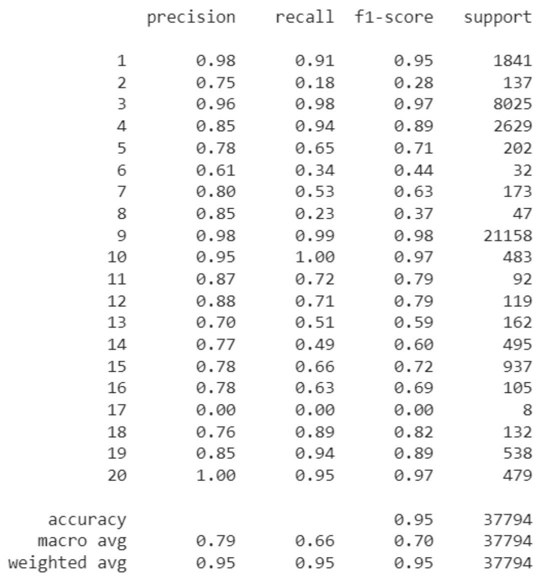


Figure 8: Predicted labels of each IoT device.



```
              precision    recall  f1-score   support

           1       0.98      0.91      0.95      1841
           2       0.75      0.18      0.28       137
           3       0.96      0.98      0.97      8025
           4       0.85      0.94      0.89      2629
           5       0.78      0.65      0.71       202
           6       0.61      0.34      0.44        32
           7       0.80      0.53      0.63       173
           8       0.85      0.23      0.37        47
           9       0.98      0.99      0.98     21158
          10       0.95      1.00      0.97       483
          11       0.87      0.72      0.79        92
          12       0.88      0.71      0.79       119
          13       0.70      0.51      0.59       162
          14       0.77      0.49      0.60       495
          15       0.78      0.66      0.72       937
          16       0.78      0.63      0.69       105
          17       0.00      0.00      0.00         8
          18       0.76      0.89      0.82       132
          19       0.85      0.94      0.89       538
          20       1.00      0.95      0.97       479

    accuracy                           0.95     37794
   macro avg       0.79      0.66      0.70     37794
weighted avg       0.95      0.95      0.95     37794
```

Figure 9: Machine learning model accuracy on Precision, Recall, and F1-Score.

Finally, we show the accuracy of all 10 IoT devices in each active and idle state with respect to Precision, Recall, and F1-Score. In order to provide a significance of the results, here we consider the captured data without balancing and show the result without modifying any values. In Figure 9, we show that the model achieves 95% of accuracy. We attribute this performance to the XGBoost model, as it performs significantly well over imbalanced classification datasets.

7 Discussion and Future Work

Our findings support the hypothesis that an outside-of-network attacker can successfully identify IoT devices without connecting to a WiFi network. The type of devices and their operating modes may be determined by characteristics like the number of packets, inter-arrival time, packet sizes, and distributions. The attack is straightforward to execute without leaving any traces or fingerprints.

The profiling attack causes serious privacy issues. A potential attacker could drive close to a business to evaluate the volume of economic activity, the socioeconomic background of the clients, expected revenues, revenue trends, or even find possible weak targets for future attacks. It can provide environmental awareness that can be utilized to track mobile devices in more complicated circumstances (e.g., cars, drones, phones, etc.). For instance, a swarm of drones can be deployed over a sizable area to classify, identify, and monitor the movement of signal-emitting devices in the covered area. This might show how gadgets communicate with one another and show where each device travels.

As we have proved that the experimental testbed is reliable and has significant importance in the real-world, the implementation in real-world scenarios is of utmost importance to secure the privacy of the individual. To this end, we aim to further extend the proposed implementation in real-world scenarios, where we would be able to show the further importance of such attacks.

8 Conclusion

This paper investigates a privacy leakage from an out-of-network eavesdropper on encrypted WiFi traffic. To this end, we consider 10 IoT devices and capture their data from outside the network without joining the WiFi network. During the investigation, we prove that IoT device eavesdropping is not only possible but also a straightforward process. To this end, we exploit the WiFi frame timing and header information and conduct a detailed evaluation using a machine learning technique for inferring and fingerprinting which IoT device exists in the network and what working status each device is. The models we found had exceptional high accuracy, they are most likely approaching the point where subsequent improvement becomes more difficult and might even come with the loss of generalizeability [27]. Our evaluation achieves high accuracy, up to 95%, in identifying the devices and their working status. The experimental results show that outside intruders can significantly harm the IoT devices without joining a WiFi network and can launch the attack within a minimum time without leaving any detectable footprints.

References

[1] M. Alyami, I. Alharbi, C. Zou, Y. Solihin, K. Ackerman, "WiFi-based IoT Devices Profiling Attack based on Eavesdropping of Encrypted WiFi Traffic," in 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), 385–392, IEEE, 2022.

[2] P. Dudhe, N. Kadam, R. Hushangabade, M. Deshmukh, "Internet of Things (IOT): An overview and its applications," in 2017 International conference on energy, communication, data analytics and soft computing (ICECDS), 2650–2653, IEEE, 2017.

[3] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. P. Costa, B. Walke, "The IEEE 802.11 universe," IEEE Communications Magazine, **48**(1), 62–70, 2010.

[4] K. Boeckl, K. Boeckl, M. Fagan, W. Fisher, N. Lefkovitz, K. N. Megas, E. Nadeau, D. G. O'Rourke, B. Piccarreta, K. Scarfone, Considerations for managing Internet of Things (IoT) cybersecurity and privacy risks, US Department of Commerce, National Institute of Standards and Technology . . . , 2019.

[5] K. Vengatesan, A. Kumar, M. Parthibhan, A. Singhal, R. Rajesh, "Analysis of Mirai botnet malware issues and its prediction methods in internet of things," in International conference on Computer Networks, Big data and IoT, 120–126, Springer, 2018.

[6] E. Ronen, A. Shamir, "Extended functionality attacks on IoT devices: The case of smart lights," in 2016 IEEE European Symposium on Security and Privacy (EuroS&P), 3–12, IEEE, 2016.

[7] A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, M. Conti, A.-R. Sadeghi, S. Uluagac, "Peek-a-boo: I see your smart home activities, even encrypted!" in Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 207–218, 2020.

[8] S. Aneja, N. Aneja, M. S. Islam, "IoT device fingerprint using deep learning," in 2018 IEEE international conference on internet of things and intelligence system (IOTAIS), 174–179, IEEE, 2018.

[9] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, N. Feamster, "Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic," arXiv preprint arXiv:1708.05044, 2017.

[10] M. W. Nadeem, H. G. Goh, M. Hussain, M. Hussain, M. A. Khan, et al., "Internet of Things for Green Building Management: A Survey," in Role of IoT in Green Energy Systems, 156–170, IGI Global, 2021.

[11] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, A. Zanella, "IoT: Internet of threats? A survey of practical security vulnerabilities in real IoT devices," IEEE Internet of Things Journal, **6**(5), 8182–8201, 2019.

[12] A. Sujanani, S. Pai, "802.11 Frame-level Network IDS for Public Wireless Networks," in ICT Systems and Sustainability, 453–461, Springer, 2021.

[13] O. Salman, I. H. Elhajj, A. Chehab, A. Kayssi, "A machine learning based framework for IoT device identification and abnormal traffic detection," Transactions on Emerging Telecommunications Technologies, **33**(3), e3743, 2022.

[14] J. S. Atkinson, J. E. Mitchell, M. Rio, G. Matich, "Your WiFi is leaking: What do your mobile apps gossip about you?" Future Generation Computer Systems, **80**, 546–557, 2018.

[15] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, V. Sivaraman, "Classifying IoT devices in smart environments using network traffic characteristics," IEEE Transactions on Mobile Computing, **18**(8), 1745–1759, 2018.

[16] I. Sanchez-Rola, I. Santos, D. Balzarotti, "Clock around the clock: Time-based device fingerprinting," in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 1502–1514, 2018.

[17] R. VanSickle, T. Abegaz, B. Payne, "Effectiveness of tools in identifying rogue access points on a wireless network," 2019.

[18] P. Serrano, M. Zink, J. Kurose, "Assessing the Fidelity of COTS 802.11 Sniffers," in IEEE INFOCOM 2009, 1089–1097, 2009, doi:10.1109/INFCOM.2009.5062021.

[19] A. Abusnaina, R. Jang, A. Khormali, D. Nyang, D. Mohaisen, "DFD: Adversarial Learning-based Approach to Defend Against Website Fingerprinting," in IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, 2459–2468, 2020, doi:10.1109/INFOCOM41043.2020.9155465.

[20] W. De la Cadena, A. Mitseva, J. Hiller, J. Pennekamp, S. Reuter, J. Filter, T. Engel, K. Wehrle, A. Panchenko, "Trafficsliver: Fighting website fingerprinting attacks with traffic splitting," in Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 1971–1985, 2020.

[21] F. Zhang, W. He, X. Liu, "Defending against traffic analysis in wireless networks through traffic reshaping," in 2011 31st International Conference on Distributed Computing Systems, 593–602, IEEE, 2011.

[22] Z. Jiang, K. Zhao, R. Li, J. Zhao, J. Du, "PHYAlert: identity spoofing attack detection and prevention for a wireless edge network," Journal of Cloud Computing, **9**(1), 1–13, 2020.

[23] Q. Xu, R. Zheng, W. Saad, Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," IEEE Communications Surveys & Tutorials, **18**(1), 94–104, 2015.

[24] I. I. A. Sulayman, R. He, M. Manka, A. Ning, A. Ouda, "LiFi/WiFi Authentication and Handover Protocols: Survey, Evaluation, and Recommendation," in 2021 International Symposium on Networks, Computers and Communications (ISNCC), 1–6, IEEE, 2021.

[25] K.-C. Lan, J. Heidemann, "On the correlation of internet flow characteristics," Technical report, Citeseer, 2003.

[26] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, et al., "Xgboost: extreme gradient boosting," R package version 0.4-2, **1**(4), 1–4, 2015.

[27] I. A. Alharbi, A. J. Almalki, C. C. Zou, "Hyperparameter Optimization and Comparison of Student Performance Prediction Algorithms," in 2021 International Conference on Computational Science and Computational Intelligence (CSCI), 889–894, 2021, doi:10.1109/CSCI54926.2021.00207.